

Hydrographic Network Extraction from Radar Satellite Images using a Hierarchical Model within a Stochastic Geometry Framework

Caroline Lacoste, Xavier Descombes, Josiane Zerubia, Nicolas Baghdadi

▶ **To cite this version:**

Caroline Lacoste, Xavier Descombes, Josiane Zerubia, Nicolas Baghdadi. Hydrographic Network Extraction from Radar Satellite Images using a Hierarchical Model within a Stochastic Geometry Framework. [Research Report] RR-5697, INRIA. 2006, pp.27. inria-00070318

HAL Id: inria-00070318

https://inria.hal.science/inria-00070318

Submitted on 19 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Hydrographic Network Extraction from Radar Satellite Images using a Hierarchical Model within a Stochastic Geometry Framework

Caroline Lacoste — Xavier Descombes — Josiane Zerubia — Nicolas Baghdadi

N° 5697

September 2005

Thème COG



ISSN 0249-6399 ISRN INRIA/RR--5697--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Hydrographic Network Extraction from Radar Satellite Images using a Hierarchical Model within a Stochastic Geometry Framework

Caroline Lacoste , Xavier Descombes , Josiane Zerubia , Nicolas Baghdadi

Thème COG — Systèmes cognitifs
Projet Ariana

Rapport de recherche n° 5697 — September 2005 — 27 pages

Abstract: This report presents a two-step algorithm for unsupervised extraction of hydrographic networks from satellite images, that exploits the tree structures of such networks. First, the thick branches of the network are detected by an efficient algorithm based on a Markov random field. Second, the line branches are extracted using a recursive algorithm based on a hierarchical model of the hydrographic network, in which the tributaries of a given river are modeled by an object process (or a marked point process) defined within the neighborhood of this river. Optimization of each point process is done via simulated annealing using a reversible jump Markov chain Monte Carlo algorithm. We obtain encouraging results in terms of omissions and overdetections on a radar satellite image.

Key-words: Stochastic geometry, marked point process, simulated annealing, RJMCMC, Markov random field, hydrographic network extraction, radar satellite image.

Extraction de Réseaux Hydrographiques à partir d'Images Satellitaires Radar par un Modèle Hiérarchique dans un cadre de Géométrie Stochastique

Résumé : Ce rapport présente un algorithme d'extraction non supervisée de réseaux hydrographiques à partir d'images satellitaires exploitant la structure arborescante de tels réseaux. L'extraction du surfacique (branches de largeur supérieure à trois pixels) est réalisée par un algorithme efficace fondé sur une modélisation par champ de Markov. Ensuite, l'extraction du linéique se fait par un algorithme récursif fondé sur un modèle hiérarchique dans lequel les affluents d'un fleuve donné sont modélisés par un processus ponctuel marqué défini dans le voisinage de ce fleuve. L'optimisation de chaque processus ponctuel est réalisée par un recuit simulé utilisant un algorithme de Monte Carlo par chaîne de Markov à sauts réversibles. Nous obtenons de bons résultats en terme d'omissions et de surdétections sur une image radar de type ERS.

Mots-clés : Géométrie stochastique, processus ponctuel marqué, recuit simulé, MCMC à sauts réversibles, champs de Markov, extraction de réseaux hydrographiques, imagerie satellitaire radar.

Contents

1 Introduction **4**

2 Data **4**

3 Segmentation using an MRF **7**
3.1 Data term . . . 7
3.2 Prior term . . . 7
3.3 Optimization . . . 7
3.4 Results . . . 8

4 Object Detection using MPPs - General framework **12**
4.1 Scene modeling using marked point processes . . . 12
4.2 Point process sampling . . . 13
4.3 Optimization . . . 13

5 Network modeling using MPPs **13**
5.1 Hierarchical modeling . . . 13
5.2 Process defined in the neighborhood of an object . . . 14
5.2.1 Reference process . . . 14
5.2.2 Prior density . . . 15
5.2.3 Data term . . . 15

6 Network extraction using a hierarchical modeling **16**
6.1 Initialization . . . 16
6.2 Generating new branches . . . 20

7 Conclusion **25**

Acknowledgments **25**

1 Introduction

Image processing is an important tool for cartographers to optimize the time spent on ground while improving the accuracy of the final map. With the availability of remotely sensed images and advances in computing technologies, many methods have been developed in order to extract cartographic items for updating geographical data. In this context, we have been interested in extracting hydrographic networks constituted of rivers and their tributaries from remotely sensed images.

Our objective is to exploit the spatial structure of these network to propose a fully-automatic technique for river extraction. Indeed, the integration of contextual information into the detection process can significantly improve the results in terms of accuracy and reliability.

In this context, Markov Random Fields (MRFs), known for their robustness with respect to noise, allow to explicitly introduce a prior knowledge on the spatial structure of the analyzed images through local conditional probabilities [24]. Nevertheless, it remains difficult to incorporate strong geometrical constraints in such models, since MRFs are defined locally. A particular class of Markov random fields, Markov random fields on graphs, allows however to incorporate strong geometrical and topological constraints into the model. They provide a powerful tool for network detection [9, 21]. The network is modeled by a graph where each arc corresponds to a possible section of the network. This is an object approach contrary to the usual MRF-based approach where the field is defined on a pixel lattice. Thus, this approach allows to introduce strong geometrical and topological constraints. A drawback of this method is that it requires an exhaustive pre-detection of possible sections: an arc not detected at the first stage will not be detected afterward.

In the same way, Marked Point Processes (MPPs), introduced in image processing in [1], allow to exploit the geometrical and topological characteristics of the networks, while benefiting from the same type of stochastic properties as those of MRFs (robustness with respect to noise, estimators, etc.). These models, also called object processes, are random sets of objects whose number is a finite random variable. Interactions between objects are taken into account in the density of the process, which allows to incorporate constraints on the network topology. This stochastic geometric framework allows to simultaneously determine the object number, the object location, and the object characteristic (length, orientation, shape, etc.) in a rigorous way. As for MRFs on graphs, it is based on the modeling of the scene by a set of objects and a global maximization of the network density (*i.e.* minimization of the network energy). The main difference with MRFs on graph concerns the fact that the object location and the object number evolve during the optimization procedure. MPPs are thus more flexible than MRFs and the pre-detection is not necessary anymore. Such models have been proposed in [15, 16, 17, 20] for unsupervised road network extraction. In [17, 20], process objects are interacting line segments. These spatial processes lead to continuous extracted line networks with few omissions and overdetections. This modeling is extended in [16] to more complex objects: the objects are interacting polylines composed by an unknown number of segments, which improve the accuracy of the detection. At the end of the algorithm, each detected polyline corresponds to the central axis of a road or a river of constant width.

In this paper, we use the same type of modeling as in [16] while exploiting the fact that the hydrographic network has a tree structure. As the manipulation of complex objects is computationally expensive [15], we propose to initialize the algorithm by an extraction of the thick branches using an MRF. Then, the line branches of the network are extracted using a recursive algorithm based on a hierarchical model of hydrographic networks, in which the tributaries of a detected river are modeled by a polyline process in the neighborhood of this river.

2 Data

The data used in this study are a radar image (ERS Satellite) over a region of French Guyana provided by the BRGM (French Geological Survey). This image is shown in figure 1. The sought-after cartographic item is the hydrographic network. This network is characterized by a tree structure, where the main river is the root of the tree and its tributaries are branches from which other branches can be generated. The radar imagery is well-adapted, as rivers correspond to dark regions in the image in a light background. The main difficulty is to extract the fine branches (width lower than 3 pixels) whose detection is perturbed by speckle noise. A manual extraction, provided by the BRGM, is given in figure 2.

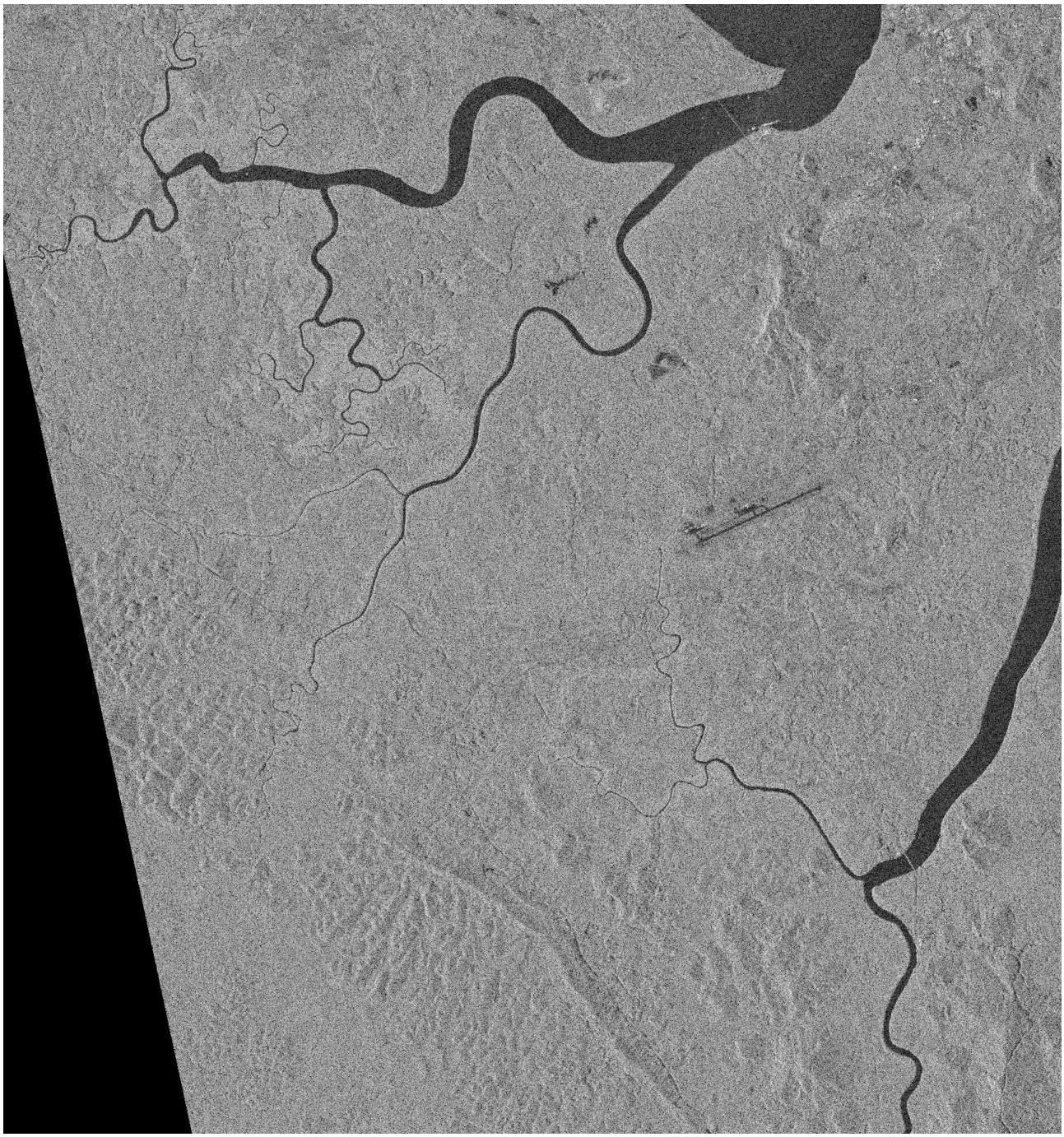


Figure 1: ERS radar image of the Guyana © BRGM. The image size is 1709×1825 pixels with a resolution of 12.5 meters.

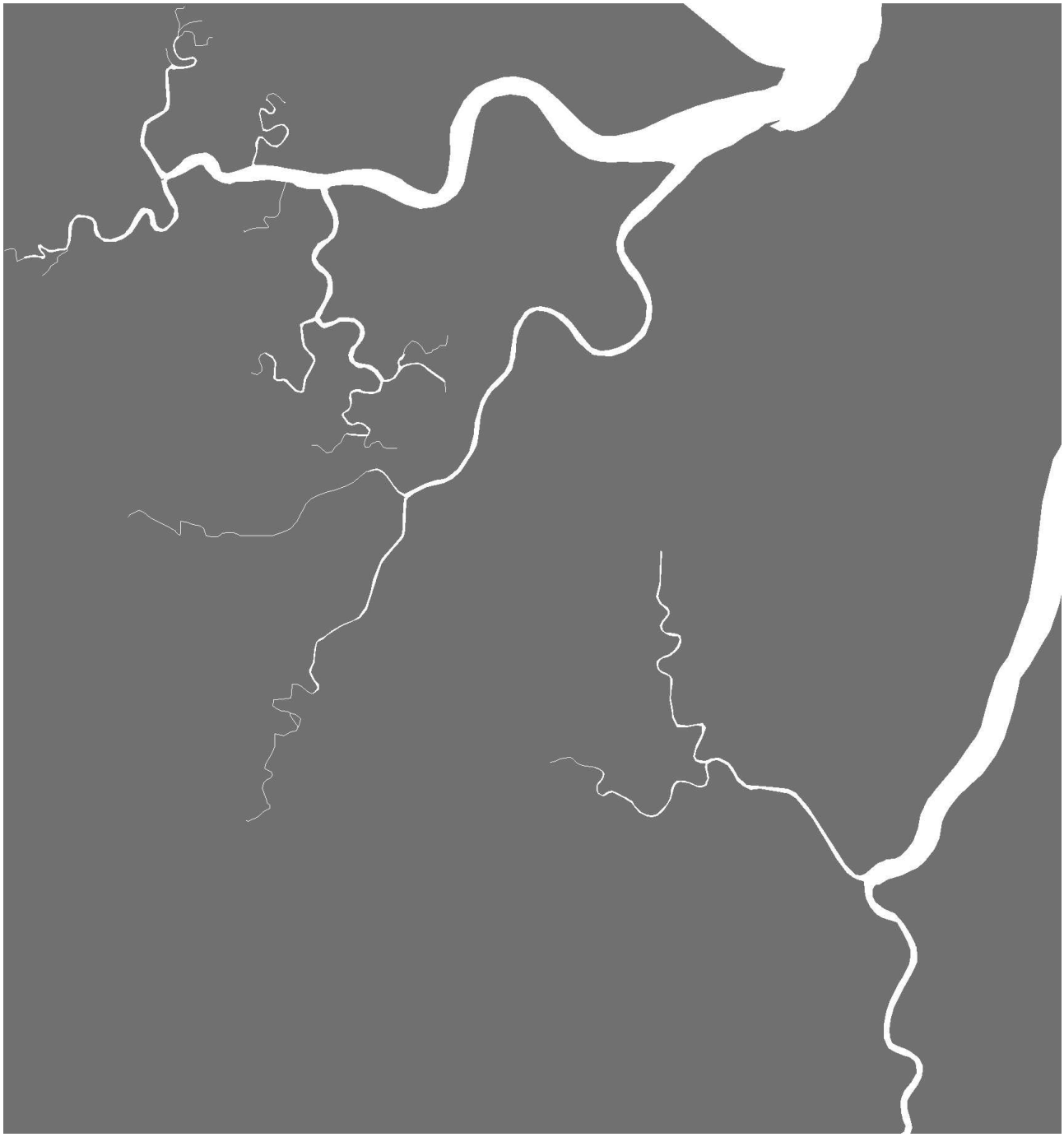


Figure 2: Manual extraction of the hydrographic network © BRGM.

3 Segmentation using an MRF

To extract the rivers from radar images, we first propose a segmentation method based on a Markov random field. We consider two labels: c_R corresponding to the rivers and c_B corresponding to the background. Given the data field Y, our goal is to find the label field X. Embedded in a Bayesian framework, a natural candidate for X is the Maximum A Posteriori (MAP) estimator:

$$\hat{X}_{MAP} = \arg\max_X P(X|Y) = \arg\min_X U(X|Y) \tag{1}$$

where the energy U can be written as follows:

$$U(X|Y) = U_d(Y|X) + U_p(X) \tag{2}$$

where U_d the data term and U_p is the prior term.

3.1 Data term

Although radar noise (speckle) is correlated, we suppose that the observation values are conditionally independent given X and that the grey level y_s only depends on the class to which it belongs, *i.e.* on the value x_s. Although erroneous, this assumption enables to simply define the observation likelihood and, above all, to gain with respect to efficiency, which is the first objective of the extraction method suggested in this section. The observation likelihood is thus given by:

$$P(Y|X) = \prod_{p \in S} l(y_s|x_s) \tag{3}$$

where $l(y_s|x_s)$ is the likelihood of y_s given x_s, *i.e.* given the parameters of the background model if $x_s = c_B$, or given the parameters of the river model if $x_s = c_R$. The chosen models consist of Gaussian distributions. The data term is then defined as follows:

$$U_d(Y|X) = \sum_s g(y_s|m_R, \sigma_R)\delta_{x_s=c_R} + g(y_s|m_B, \sigma_B)\delta_{x_s=c_B} \tag{4}$$

where $g(.|m, \sigma)$ is the Gaussian log-likelihood function of mean m and standard deviation σ, m_R and σ_R (resp. m_B and σ_B) correspond to the empirical mean and variance of the pixels whose label is c_R (resp. c_B). The values m_R, σ_R, m_B and σ_B are updated during the optimization algorithm at each scanning of the image.

Tests have been performed with an assumption of multiplicative noise, theoretically more adapted to radar imagery. The best results were nevertheless obtained with an additive Gaussian noise.

3.2 Prior term

To regularize the classification while preserving the edges we define a boolean line process as proposed by [8] for image restoration. This process explicitly represents the presence of an edge in the image. The prior term is then given by:

$$U_p(X) = \beta \sum_{<s,t>} \delta_{x_s \neq x_t} \, (1 - b_{<s,t>}) \tag{5}$$

where β is a positive weigh, $< s, t >$ denotes the pair of neighboring pixels s and t, x_s is the value of X at pixel s, $b_{<s,t>}$ denotes the value of the line field B between the pixels s and t, and δ_A is equal to 1 if A is true and 0 if not. In order to be efficient, we supposed that the realization of the line process is known. We use a Canny-Deriche filter followed by a hysteresis thresholding to compute the line field [5, 6]. This filter is applied after a median filter to reduce the effect of the speckle noise.

3.3 Optimization

A natural choice for optimization could be a simulated annealing using a Gibbs sampler. Although this method leads to the MAP estimate (with an adequate temperature decrease), it is a very computationally demanding method. A faster alternative is to use a deterministic relaxation method, for instance Iterative Conditional Mode

(ICM) [3]. This algorithm converges to a local optimum located in the neighborhood of the initial configuration. The initialization is thus crucial. In addition of the estimation of the hidden field X, the model parameters $\theta = \{m_R, \sigma_R, m_B, \sigma_B\}$ are iteratively estimated during the ICM by the maximum likelihood estimator (given the label field estimated at the previous iteration). Notice that we have chosen not to estimate β (weight of the prior term) but to fix it according to the expected regularization effect.

Given that the river intensity is distinctly lower than the background intensity, we choose the following initial values for the model parameters: $m_B = \sigma_B^2 = 160 >> m_R = \sigma_R^2 = 100$. The initial label field is then estimated by a maximum likelihood approach [2], which provides an initial configuration close to the optimal configuration. The optimization algorithm is given in table 1.

1. **Initialization**:

   $\hat{\theta}_0$ empirically chosen.

   $\hat{X}_0$ estimated by maximum likelihood:

$$x_s = \begin{cases} c_B & \text{if } g(y_s|\widehat{m_B}, \widehat{\sigma_B}) > g(y_s|\widehat{m_R}, \widehat{\sigma_R}) \\ c_R & \text{if not} \end{cases}$$

2. **Parameter estimation**: $\hat{\theta}_n = \arg\max_{\theta} P(Y|X_{n-1}, \theta)$

$$n_B = \sum_{s \in S} \delta_{x_s = c_B} \quad , \quad n_R = \sum_{s \in S} \delta_{x_s = c_R}$$

$$\widehat{m_B} = \frac{\sum_{s \in S} y_s \, \delta_{x_s = c_B}}{n_B} \quad , \quad \widehat{m_R} = \frac{\sum_{s \in S} y_s \, \delta_{x_s = c_R}}{n_R}$$

$$\widehat{\sigma_B}^2 = \frac{(\sum_{s \in S} y_s^2 \, \delta_{x_s = c_B}) - n_B \, \widehat{m_B}^2}{n_B - 1} \quad , \quad \widehat{\sigma_R}^2 = \frac{(\sum_{s \in S} y_s^2 \, \delta_{x_s = c_R}) - n_R \, \widehat{m_R}^2}{n_R - 1}$$

3. If $\hat{\theta}_n = \hat{\theta}_{n-1} = \hat{\theta}_{n-2}$, **stop**.

4. **Estimation of the label field** performing one iteration of ICM:

   For each pixel s, computation of the conditional energy difference:

$$\Delta U(s) = U(x_s = c_B \,|\, y_s, \{x_t\}_{t \in N_s}, \hat{\theta}_n) - U(x_s = c_R \,|\, y_s, \{x_t\}_{t \in N_s}, \hat{\theta}_n)$$

   If $\Delta U(s) < 0$, $x_s = c_B$, else $x_s = c_R$.

5. If the maximal number of iterations is reached, **stop**. Else, return to 2.

Table 1: ICM Algorithm.

3.4 Results

Tests have been performed on an original image and an image filtered by a Gaussian kernel. The obtained results are slightly better for the filtered image. Moreover, the ICM convergence was detected in less than 10 seconds for a filtered image of size 911×853 and in 26 seconds for the original image of size 1709×1825 (with a Pentium III processor, 1 GHz, 1 Go of RAM). The figure 3 shows the results obtained on the filtered image. All the thick branches (larger than 3 pixels) are detected. The few false alarms can be easily removed by a morphological post-processing as shown in figure 4. Nevertheless, the fine branches (lower than 3 pixels) are not detected.

We have also tested a more elaborate MRF model, called the "chien-model", preserving not only edges but also lines. This model was introduced in [7] for binary image restoration. A result using this model and a simulated annealing is given in figure 5. Despite some improvements, a large part of the lines of the network is still omitted.

This study shows the limits of MRFs that only take into account local interactions. This motivates the use of MPPs that allow incorporation of strong geometrical constraints. Nevertheless, the MPP approaches are computationally expensive, and all the more when the manipulated objects are complex [15]. This is the reason why we propose a two-steps algorithm where the surface part of the network is detected by MRFs and the line part of the network is then extracted by MPPs where the objects are reduced to polylines.

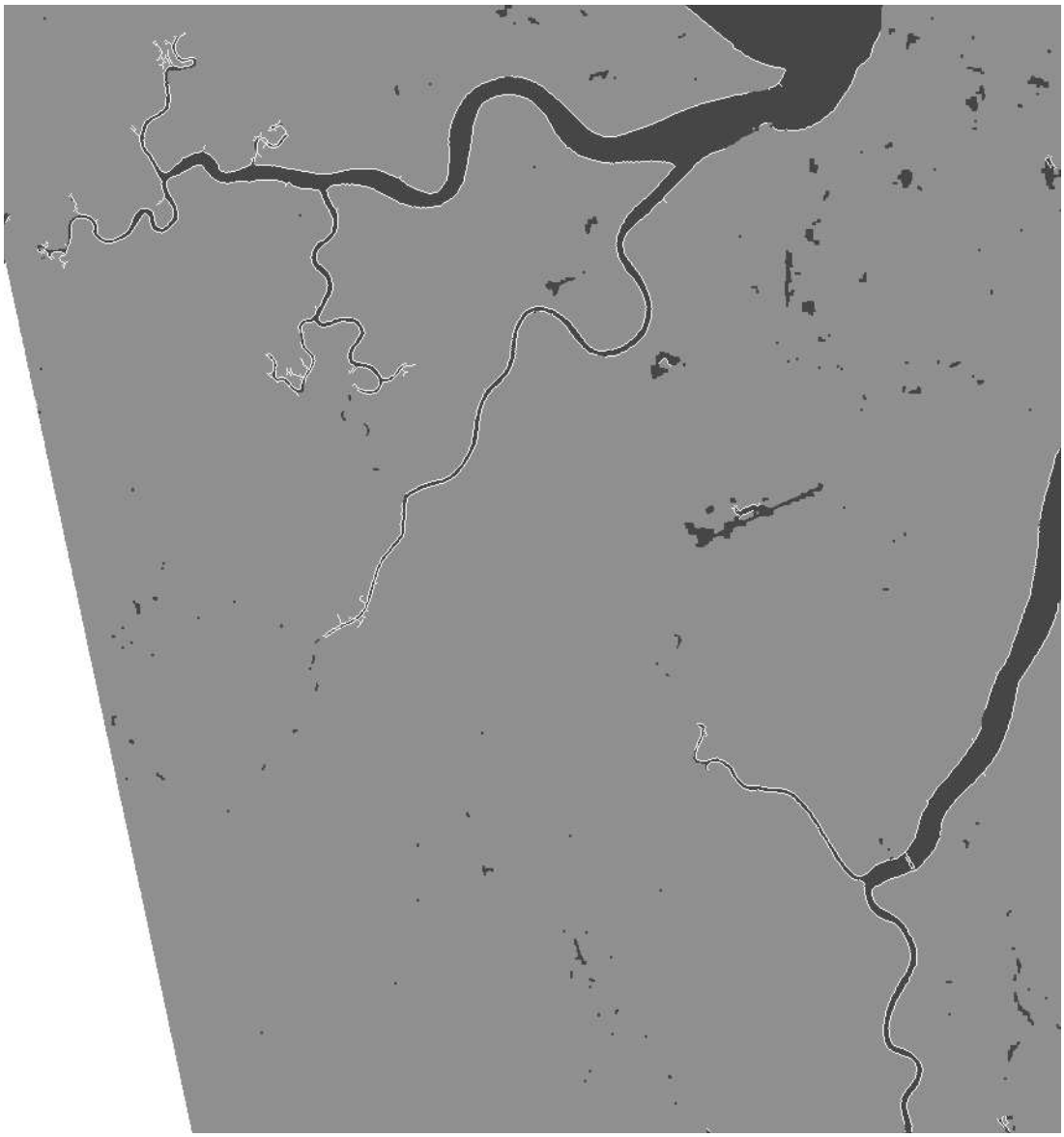


Figure 3: Classification using ICM - BLACK = rivers - GREY = background - WHITE = edges obtained by a Canny-Deriche filter followed by a hysteresis thresholding.

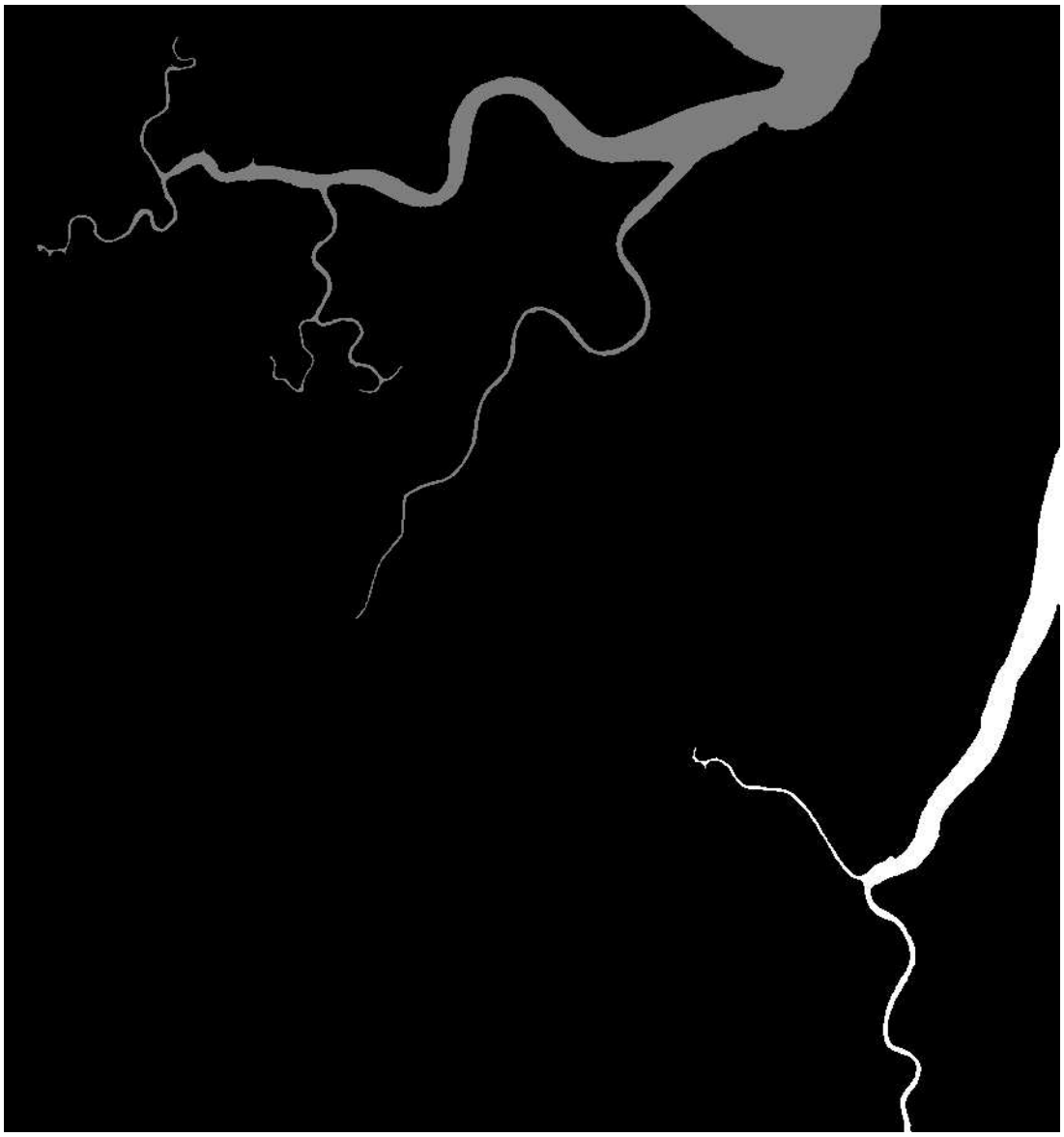


Figure 4: Post-processing: morphological closing by a line structuring element (six pixel long, with four orientations), followed by a connected component extraction (8-connectivity), and then by the removing of small components. The first component is grey. The second is white.

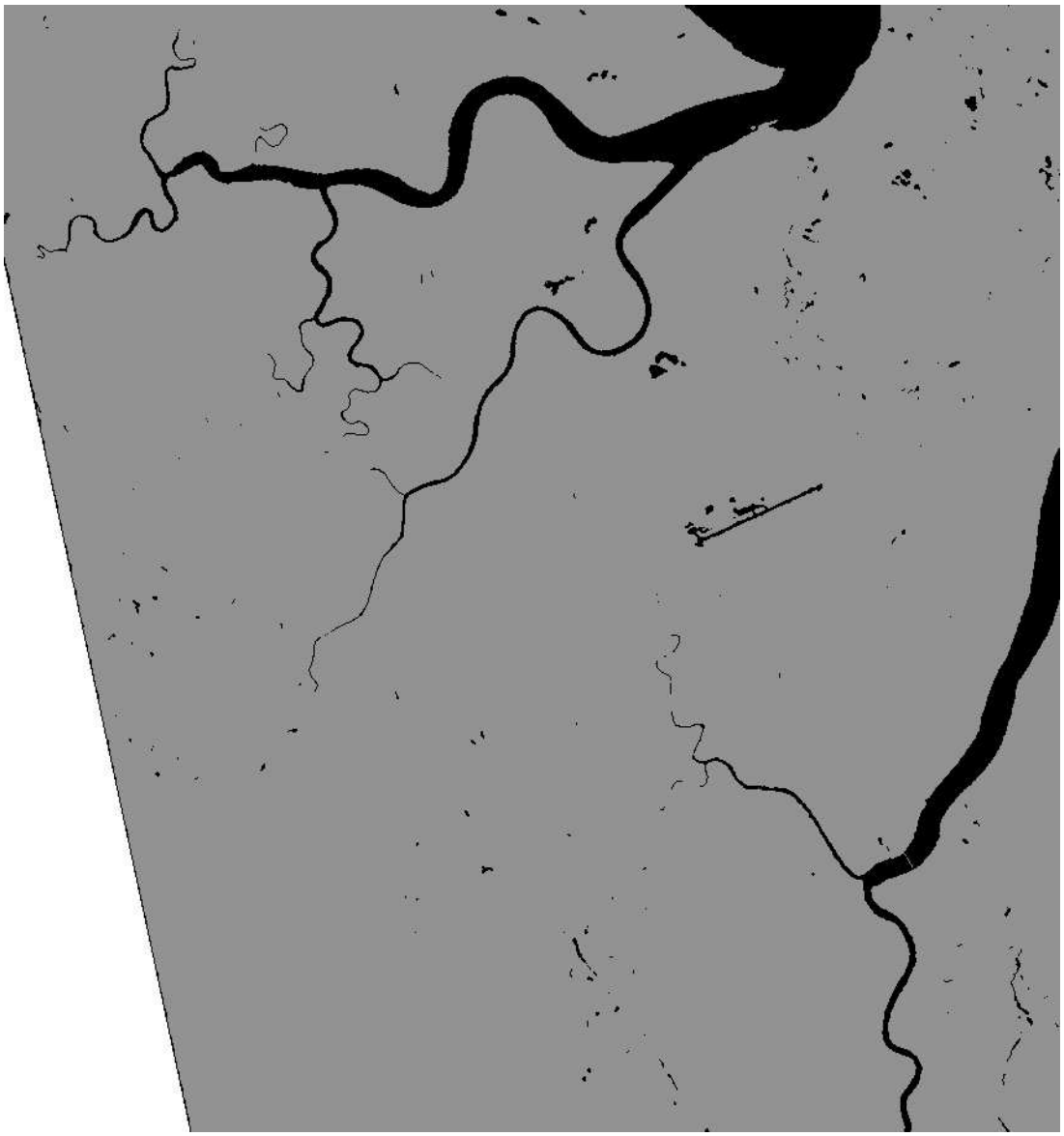


Figure 5: Classification obtained by simulated annealing with a Gibbs sampling using the "chien-model" as prior model.

4 Object Detection using MPPs - General framework

4.1 Scene modeling using marked point processes

Point processes provide a rigorous framework based on measure theory to describe a scene by an unordered set of points in a compact set $F \subset \mathbb{R}^d$ [23]. For $n \in \mathbb{N}$, let Ω_n be the set of configurations $\{x_1, \ldots, x_n\}$ that consist of n unordered points of F. A **point process** on F is a mapping $\mathbf{X}$ from a probability space to the set of configurations $\Omega = \bigcup_{n=0}^{\infty} \Omega_n$, such that for all bounded Borel set $A \subseteq F$, the number of points $N_{\mathbf{X}}(A)$ falling in A is a finite random variable.

The canonical "completely random" point process is the **uniform Poisson point process**. Under the law of a Poisson point process of intensity λ, the number of points $N_{\mathbf{X}}(A)$ follows a Poisson law with mean $\lambda|A|$, and, given $N_{\mathbf{X}}(A) = n$, the n points are independently and uniformly distributed in A. The law of a Poisson process of intensity λ on $F \subset \mathbb{R}^d$ is defined by the following probability measure on $(\Omega, \mathcal{B})$[1]:

$$\mu(B) = \sum_{N=0}^{\infty} \frac{\lambda^N e^{-\lambda|F|}}{N!} \int_{F^N} \mathbf{1}_B(\{x_1, \ldots, x_N\})\, dx_1 \ldots dx_N \tag{6}$$

where $B \in \mathcal{B}$.

To model the observed scene by a set of objects, we can augment a point process by adding extra information (*i.e.* object parameters) to each points. Such a process is called a **marked point process** or an **object process**. A marked point process on F, with marks in a space M, is a point process on $F \times M$ such that $N(A \times M) < \infty$ almost surely for any bounded Borel set $A \subset F$. In this context, the uniform Poisson process is a marked point process where points are distributed according to a uniform Poisson point process of intensity λ, and marks associated to each point are uniformly distributed in M. The law of this process on $F \subset \mathbb{R}^d$ is defined by the following probability measure:

$$\mu(B) = \sum_{N=0}^{\infty} \frac{e^{-\lambda F}}{N!} \int_{(F \times M)^N} \mathbf{1}_B(\{c_1, \ldots, c_N\})\, dx_1 dP_M(m_1) \ldots dx_n dP_M(m_N) \tag{7}$$

where the $c_i = (x_i, m_i)$ are marked points of $F \times M$, and P_M is the uniform probability measure on the mark space M.

Although in most applications it is not realistic to assume that points are scattered randomly, Poisson processes are useful to build more complex models. Indeed, interactions can be introduced by specifying a **density** with respect to the reference measure μ. Let h be a nonnegative function on Ω. Then, the measure ν having a density h with respect to μ is defined by:

$$\nu(B) = \int_B h(C)\, \mu(dC) \tag{8}$$

If $0 < \nu(B) < \infty$, then ν can be normalized to provide a probability measure π defined by: $\nu(B)/\nu(\Omega)$. Interaction models are usually specified by an unnormalized **Gibbs** density given by:

$$h(X) = \exp\left(-U(X)\right) \tag{9}$$

where $U(X)$ is the **energy** of the system. In the context of object extraction from images, the density is usually factorized into two terms. First, geometrical and topological constraints are incorporated through a **prior density** h_p. Second, a **data term** h_d is used to fit the data. The complete density of the spatial process is then defined as follows:

$$h(C) \propto h_d(C)\, h_p(C) \tag{10}$$

where $C = \{c_1, \ldots, c_n\}$ is a configuration of objects.

[1] $\mathcal{B}$ is the smallest σ-algebra such that for all Borel sets $A \subseteq F$ the mapping $\{x_1, \ldots, x_n\} \mapsto N_{\mathbf{X}}(A)$ is measurable.

4.2 Point process sampling

Let us recall that the distribution π of a marked point process is supported by a state space of varying dimension Ω arising from an unknown number of objects. To deal with this unnormalized probability measure in which the random variable has a randomly variable dimension, two types of algorithms have been proposed in the literature : **Birth and Death (BD) samplers**, based on a spatial birth and death processes [18, 19], and **Reversible Jump Monte Carlo Markov Chain (RJMCMC) algorithm** with a Metropolis-Hastings-Green dynamics [10, 11, 12]. Two main reasons have prompted us to use a RJMCMC algorithm to simulate the model defined in the previous section. First, in many cases, the BD sampler is less efficient, as it has been shown in [14]. Second, the BD algorithm only allows birth or death of an object, whereas several types of updates - such as translation, rotation, dilation of an object - can be defined in the RJMCMC algorithm. So, if relevant moves are proposed, the algorithm convergence is faster, especially when the process is geometrically constrained.

The RJMCMC algorithm consists in simulating a discrete Markov chain C_t which performs small jumps between the spaces Ω_i. The measure of interest occurs as the stationary measure of the chain. This iterative algorithm does not depend on the initial state. At each step, a transition from the current state C to a X-Mozilla-Status: 8000 X-Mozilla-Status2: 00000000 new state C' is proposed according to a proposal kernel $Q(C \rightarrow .)$. The transition is accepted with a probability $\alpha(C, C')$ given by the Green ratio. This acceptance ratio is computed so that the detailed balance condition is verified, condition under which this algorithm converges to π. The proposal kernel Q can be decomposed into several kernels q_i, each corresponding to a reversible move, as it has been proposed in [12].

4.3 Optimization

To extract objects from an image, we aim to find a configuration $\widehat{C}$ which maximizes the unnormalized process density with respect to the reference process measure. This is a non convex problem for which a direct optimization is not possible given the large size of the state space $\Omega = \bigcup_{n=0}^{\infty} \Omega_n$, where Ω_n is the set of configurations of n objects. We propose to estimate this maximum by a **simulated annealing**, which consists of successive simulations of the process distribution π_T specified by the density $h^{1/T}$, with the temperature T gradually decreasing to 0. A proof of convergence towards the global maximum is given in [22] when the decrease of temperature T is logarithmic. In practice, the temperature decreases faster in order to reduce the computing time. The most commonly used decreasing law is the geometric decrease:

$$T(t+1) = cT(t)$$

where c is a constant close to one. A more relevant choice consists in using an **adaptive temperature decrease** which allows to decrease as much as possible the temperature while remaining close to the balance. An implementation of this idea is given in [13]. It consists of implementing a geometric decrease in stages where the stage lengths depend on the variations of the energy empirical mean. Notice that if the equilibrium is reached, this mean is supposed to fluctuate around the true value. The energy empirical mean is computed on regular intervals of length N, for which the temperature is constant. For the range $i = [t_i, t_i + N]$, the energy mean is given by:

$$\langle U \rangle_i = \frac{1}{N} \sum_{k=t_i}^{t_i+N} U(C_k) \tag{11}$$

At the end of an interval i, the energy mean over this period is compared to the one of the previous interval $i-1$. The temperature decreases at the next period only if there is an increase of the energy mean:

$$T_i = \begin{cases} T_{i-1} & \text{if} \quad \langle U \rangle_{i+1} \leq \langle U \rangle_i \\ cT_{i-1} & \text{if} \quad \langle U \rangle_{i+1} > \langle U \rangle_i \end{cases} \tag{12}$$

5 Network modeling using MPPs

5.1 Hierarchical modeling

In this section, we model the network by a collection of objects having a hierarchical structure, each object corresponding to a river. The first level of the hierarchy represents the main rivers of the observed scene and can be modeled as a marked point process in the observed frame. For each object of the first level a process

is defined in its neighborhood to model the tributaries of the corresponding main river. For each tributary, a process is defined in its neighborhood to model its tributaries, and so on.

This tree structure allows to generate each river (level 1), its tributaries (level 2), the tributaries of these tributaries (level 3), and so on, in a recursive way. Nevertheless, the computing time necessary to obtain the first level would be prohibitive. Indeed, the manipulation of large objects induces an important computing time at each proposition of perturbations, the data term being computed on numerous pixels. A possible approach could be to work in a multi-resolution framework. Here, we propose to use the segmentation by an MRF presented in section 3 that provides the surface part of the network. Consequently, we consider all branches larger than three pixels as known in the next sections.

5.2 Process defined in the neighborhood of an object

5.2.1 Reference process

Let $\mathbf{C}$ be the set of detected objects. Each object $c \in \mathbf{C}$ is described by a polyline corresponding to its central axis and its projection in the image $S(c)$. Let E_c be the equivalent in the continuous domain of $S(c)$. E_c is thus defined as a bounded set of $\mathbb{R}^2$ which is delimited by the edges of the object c.

For each object $c \in \mathbf{C}$, we defined a reference object process within the influence zone $V(c) \subset \mathbb{R}^2$ of c, defined as follows:

$$p \in V(c) \Leftrightarrow \begin{cases} d(p,c) < d_{max} \\ p \notin E_{\mathbf{C}} = \bigcup\limits_{c \in \mathbf{C}} E_c \\ c = \arg\min\limits_{\mathbf{C}} d(p,c) \end{cases} \tag{13}$$

where $d(p,c)$ denotes the Euclidean distance between p and the edges of c. The figure 6 illustrates this definition.

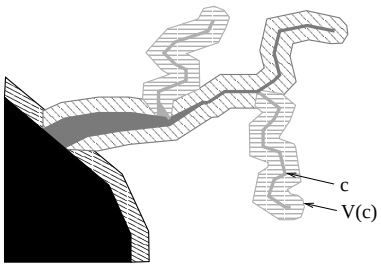


Figure 6: Influence zones.

Each object of the reference process is a polyline described by:

- an initial point $p \in V(c)$;
- an unknown number n of segments;
- the segment lengths $l_j \in [L_{min}, L_{max}]$, $j = 1, \ldots, n$;
- the segment directions:
  - $\alpha_1 \in]\alpha_0 - \pi/2, \alpha_0 + \pi/2]$ where α_0 is the direction of the line perpendicular to the central axis of c that contains p;
  - $\alpha_j \in]-\pi, \pi]$, $j = 2, \ldots, n$.

Under the reference process law, the number of polylines follows a Poisson law of mean $\lambda|V(c)|$, the initial points are uniformly distributed in $V(c)$ and the other parameters are independently and uniformly distributed in their respective state space.[2]

[2]The specific generation of the initial direction allows to avoid the generation of tributaries inside the river.

5.2.2 Prior density

To introduce an *a priori* on polyline shapes and interactions between polylines, we then specify the process by a prior density h_p with respect to the reference process law. The expression of h_p is the following:

$$h_p(X_c|\mathbf{C}) = \exp(-\sum_{x \in X_c} u_p(x|\mathbf{C} \cup X_c \setminus x)\ ) \tag{14}$$

where X_c is the configuration of objects defined with respect to $\mathbf{C}$ and u_p is given by:

$$u_p(x|\mathbf{C} \cup X_c \setminus x) = \begin{cases} +\infty \text{ if } \exists s \in X_c : |S(s) \cap S(\mathbf{C} \cup X_c \setminus s)| > \dfrac{|S(s)|}{2} \\ u_{p1}(n) + \displaystyle\sum_{j=0}^{n-1} u_{p2}(\alpha_j, \alpha_{j+1}) \text{ if not} \end{cases} \tag{15}$$

The prior term u_p **forbids the overlapping** of more than 50% of the area $S(s)$ covered by a segment s of a polyline with the aera $S(\mathbf{C} \cup X_c \setminus s)$ covered by the rest of the network. It **favors long polyline** through a potential u_{p1} on the number of segments n composing a polyline x:

$$u_{p1}(n) = \frac{M_n}{(n+1)^2} \tag{16}$$

where M_n is a positive weight. Moreover, u_p **favors slight curvature** and **forbids too acute angles** through a potential u_{p2} on pairs of successive orientations $\{\alpha_j, \alpha_{j+1}\}$:

$$U_{p2}(\alpha_j, \alpha_{j+1}) = \begin{cases} +\infty \text{ if } \cos(\alpha_j - \alpha_{j-1}) < -0.8 \\ M_\alpha \ \left(\frac{1}{2} - \cos(\alpha_{j+1} - \alpha_j)\right) \text{ if not} \end{cases} \tag{17}$$

where M_α is a positive weight.

5.2.3 Data term

The incorporation of data properties is done via a data term h_d based on a local contrast measure of the projection of the current configuration $S(X_c)$ in the image with its nearby background.

To compute the contrast value, we associate to each segment s composing the polylines a mask of pixels $M_s = (S, B)$ composed of:

- an internal region S corresponding to the object in the image;
- an external region B corresponding to the nearby background.

S is composed of the discrete segment - computed by the Bresenham method [4] - and the neighboring pixels in the normal direction with the value v satisfying:

$$g(v|m_R, \sigma_R) > g(v|m_B, \sigma_B) \tag{18}$$

where $g(.|m, \sigma)$ is the Gaussian log-likelihood function, m_R and σ_R (resp. m_B and σ_B) correspond to the grey level empirical mean and variance of the rivers (resp. background) computed on the result of the MRF-based segmentation. The other pixels of the mask are assigned to B. The line width (supposed to be lower or equal than 3 pixels) is thus implicitly taken into account through observations. This **adaptive mask** is illustrated in figure 7.

The contrast between S and B is evaluated using the statistical measure generally used to perform Student t-test, which allows to evaluate if the means of two sets are significantly different. Moreover, the river radiometry is supposed to be lower than the radiometry of the background. The contrast value is consequently nullified if $\overline{S}$, the grey level mean on S, is larger than $\overline{B}$, the grey level mean on B. Finally, the **local contrast measure** associated to the mask (S, B) is given by:

$$v(S, F) = \begin{cases} \dfrac{\overline{B} - \overline{S}}{\sqrt{\frac{\sigma_S^2}{n_S} + \frac{\sigma_B^2}{n_B}}} & \text{if } \overline{S} < \overline{B} \\ 0 & \text{if not} \end{cases} \tag{19}$$

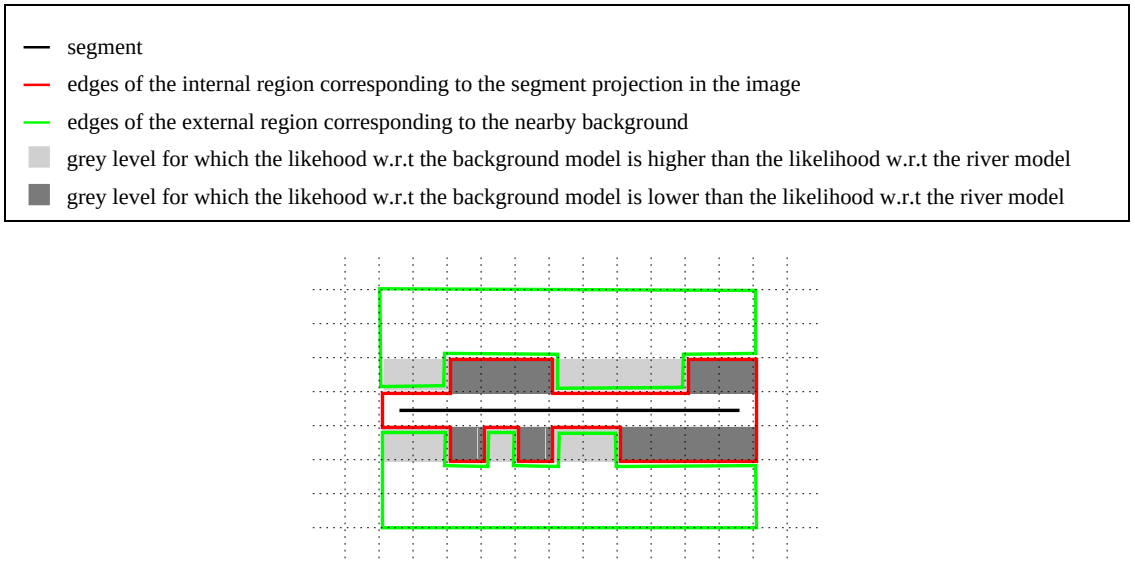


Figure 7: Adaptive mask associated to each segment composing configuration polylines.

where σ_x and n_x refer to the sample standard deviation and the number of observations of x (for $x = S, B$). Let $M(X_c)$ be the set of pixels belonging to the masks of the segments of the configuration X_c. Each pixel $p \in M(X_c)$ belongs to at least one mask. For each mask M that includes p, we have computed a contrast value v_M. The local contrast value at pixel p is then the minimal contrast value computed on these masks:

$$v_c(p) = \min_{M/p \in M} v_M(p) \tag{20}$$

Finally, the data term is given by:

$$h_d(X_c|\mathbf{C}) = \exp\left(- \sum_{p \in M(X_c)} u_c(p|\mathbf{C}) \right) \tag{21}$$

where $u_c(p)$ is a potential directly based on the local contrast measure $v_c(p)$:

$$u_c(p|\mathbf{C}) = \begin{cases} 2 & \text{if } v_c(p) < \tau_1 \text{ or } p \in S(\mathbf{C}) \\ 1 - 2\,\dfrac{v_c(p) - \tau_1}{\tau_2 - \tau_1} & \text{if } \tau_1 \leq v_c(p) \leq \tau_2 \text{ and } p \notin S(\mathbf{C}) \\ -1 & \text{if } v_c(p) > \tau_2 \text{ and } p \notin S(\mathbf{C}) \end{cases} \tag{22}$$

where τ_1 and τ_2 are positive thresholds. For river extraction from the image in figure 1, we have empirically chosen the following values: $\tau_1 = 4$ and $\tau_2 = 8$.

6 Network extraction using a hierarchical modeling

We present in this section our algorithm for hydrographic network extraction based on the hierarchical modeling proposed in section 5. The surface part of the network (*i.e.* branches larger than three pixels) is quickly obtained by an MRF-based segmentation followed by a post-processing used to obtain an object representation of the scene. This first step is described in section 6.1. Then, the line network extraction is done via a recursive algorithm based on the definition of polyline processes with respect to the previously detected branches. This generation phase is described in section 6.2.

6.1 Initialization

Step 1: From pixels to objects

The network initialization is based on the segmentation using the MRF presented in section 3. The morphological post-processing provides a connected component for each network composed of a main river, its tributaries, the

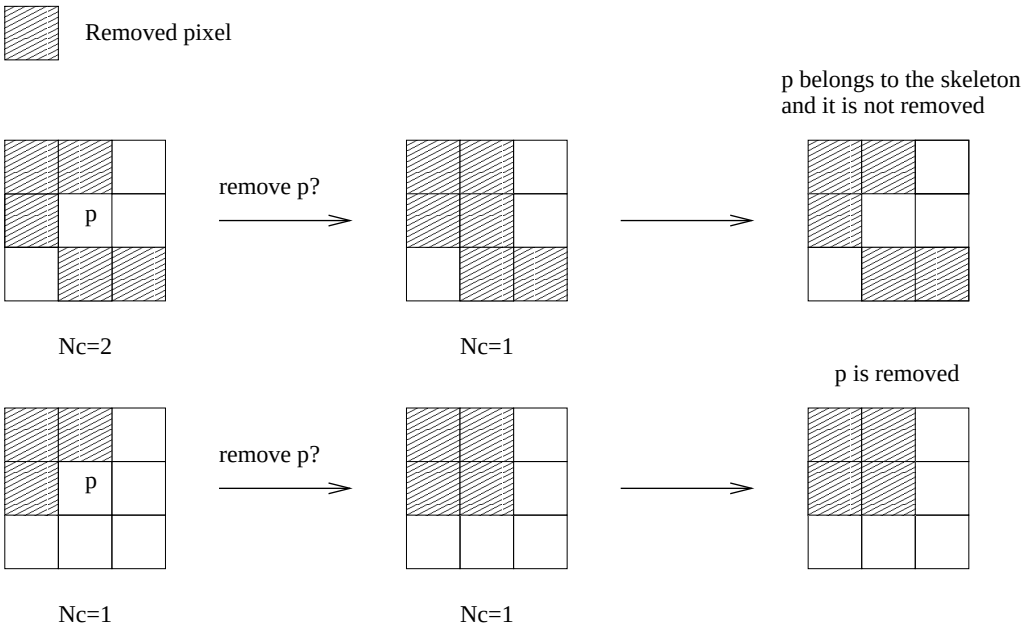


Figure 8: Removing of pixel p according to the evolution of the number of connected components composed of removed pixels.

tributaries of these tributaries, etc. To go from pixels to objects, we propose a two step algorithm which consists of:

1. **Extraction of the skeleton of each connected component** The skeleton of an object corresponds to the crest lines of the hypersurface D defined by the Hausdorff distance to the object edges, and computed for each pixel of the considered object (*i.e.*of the connected component). This distance is computed by a distance propagation in 4-connectivity from the object edges. The skeleton is then obtained by successive thinning removing pixels that do not belong to a crest line at each distance level $D = 0, 1, 2$, etc. For testing if a pixel p belongs to a crest line, we analyze the frame of size 3×3 centered in p. Let Nc be the number of connected components composed of pixels removed in the previous steps. If the removing of p induces a decrease of Nc, then p belongs to the crest line, else p is removed. This is illustrated by figure 8. Notice that this procedure does not work for final pixels (the leaves of tree) since there is only one component connect. Before the thinning procedure, the pixels belonging to the terminal line of width one are thus added to the skeleton. The algorithm result is given in figure 9.

2. **Vectorization of the skeleton to obtain a tree of polylines:**

   We first convert the skeleton to a binary tree of pixel chains. The tree root is initialized by the culminating point of the hypersurface D. The pixels are then chained until the first detected junction point. From this point, the root is the first chain and the two descendants of this root are initialized by the two possible directions. A recursive procedure provides the whole tree.

   Second, each chain is vectorized in a polyline. A width, provided by the distance to the edges D, is assigned to each checking point.

   Finally, the binary tree is converted into a n-ary tree in the following way. The main branch (*i.e.* root of the n-ary tree) is initialized by the vector version of the binary tree root, to which is concatenated:

   - D_1, its closer descendant with respect to the width and the orientation difference between the final vector of the ascendant and the initial vector of the descendant;
   - D_2, the closer descendant of D_1;
   - and so on until reaching one of the leaves of the binary tree.

   Each not linked descendant corresponds to the beginning of a n-ary tree root descendant. Each descendant is then extended in the same way as the root of the binary tree.

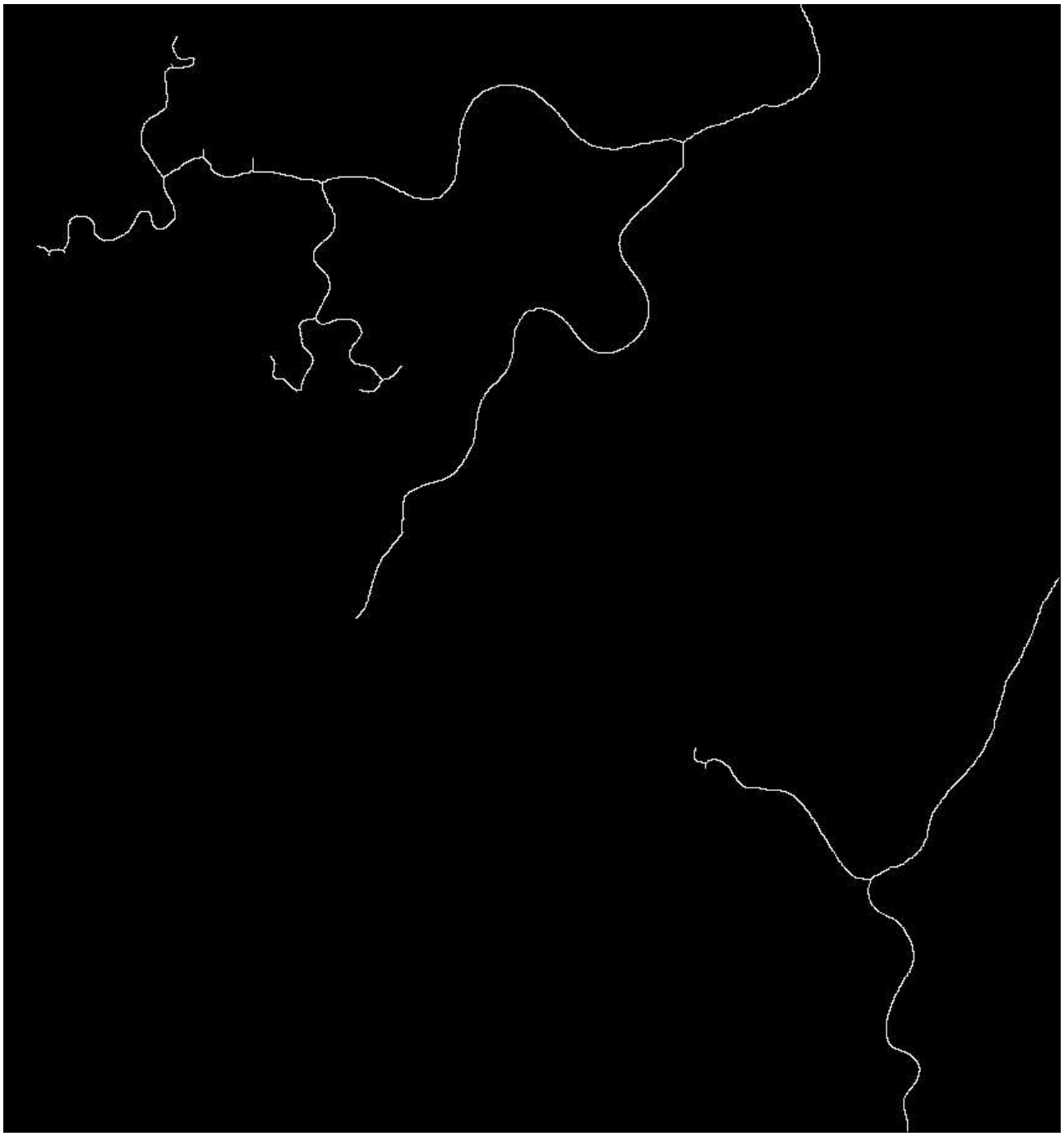


Figure 9: Skeleton of the connected components presented in figure 4.

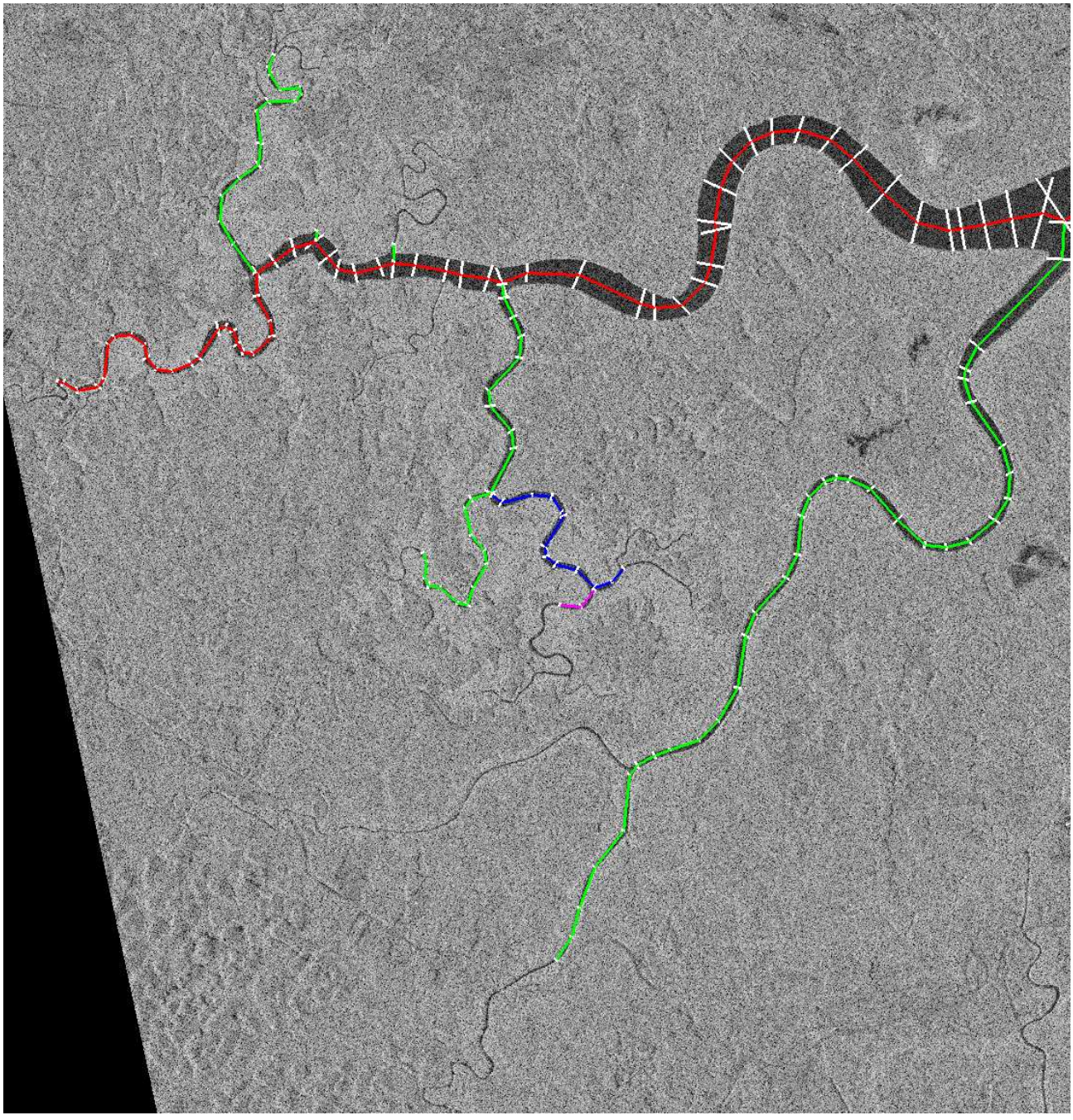


Figure 10: n-ary tree representing the first connected component (in white in figure 4). The red line corresponds to the tree root; the green lines correspond to the descendants of the root; the blue lines to the descendants of the green lines; the violet ones to the descendants of the blue ones. The white lines represent the object widths at each checking point.

Step 2: branch extension using simulated annealing

This first step provides thus a tree of objects corresponding to the surface part of the network. We have then a partial representation of the detected objects: as it is shown in figure 10, the ends of branches are omitted as the river width decreases in the direction of the spring. To extend each polyline x, we propose to estimate the set of the final parameters $\hat{v}$ (orientations and lengths of the final segments) which minimizes the energy

associated to the extended polyline $x_v = (x, v)$:

$$\widehat{v} = \arg\min_{v}[u_p(x_v|\mathbf{C}) + \sum_{p \in M(x_v)} u_c(p|\mathbf{C})] \tag{23}$$

where u_p and u_c are respectively given by equation (15) and (22). As the number of final parameters is unknown, the optimization is done via a **simulated annealing** using an **Reversible Jump Monte Carlo Markov Chain (RJMCMC)** algorithm with a Metropolis-Hastings-Green dynamics [10, 11, 12]. We use the adaptive temperature decrease proposed in section 3.3 and an automatic convergence detection based on the difference of successive energy empirical means. The perturbations proposed in the sampling algorithm do not modify the initial parameters of the considered polyline. We use two reversible moves: add/remove (AR) a segment to/from the end of the polyline; translation (T) of a point of the polyline.

The **add/remove perturbation** consists of the following steps. First, the choice of a move type is done according to a probability based on the number of segments composing x_v. If the number n of segments of x_v is equal to the initial segment number n_x, only the addition of a segment will be proposed. Likewise, if the extended polyline is composed of n_{max} segments, only the removal of a segment will be proposed. In the other cases, the removal and the addition of a segment are proposed with probability $1/2$. Second, if the addition is chosen, a length l and a direction α are uniformly drawn in $V = [L_{min}, L_{max}]\times] - \pi, \pi]$ and are added to the final parameters v. If the removal is chosen, the two last parameters l_n and α_n are removed from v. In cases where the segment number n is between $n_x + 1$ and $n_{max} - 1$, the Green ratio is given by the ratio of the densities:

$$R_{\mathrm{AR}}(x_v, x'_v) = \frac{h(x'_v|\mathbf{C})}{h(x_v|\mathbf{C})} \tag{24}$$

where x'_v is the configuration obtained by the perturbation of the current polyline x_v. In the other cases, the Green ratio is given by:

$$R_{\mathrm{AR}}(x_v, x'_v) = \frac{h(x'_v|\mathbf{C})}{2h(x_v|\mathbf{C})} \tag{25}$$

The constant $1/2$ intervenes as the addition of a segment to a polyline composed of n_x segments (resp. the removal of a segment from a polyline containing n_{max} segments) is done with probability 1 and the inverse move, *i.e.* the removal of a segment of a polyline composed of $n_x + 1$ segments (resp. the addition of a segment to a polyline composed of $n_{max} - 1$ segments) is chosen with probability $1/2$.

The **translation perturbation** consists of uniformly choosing a checking point among the new checking points (from n_x+2 to $n+1$) and proposing a translation of this checking point. This translation is a symmetrical transformation parameterized by a vector uniformly drawn in a compact centered to the origin. The Green ratio is then reduced to the ratio of the densities:

$$R_{\mathrm{T}}(x_v, x'_v) = \frac{h(x'_v|\mathbf{C})}{h(x_v|\mathbf{C})} \tag{26}$$

The figures 11 and 12 respectively show the algorithm result on each tree obtained by the first step. The extension result of the eight branches of the first tree was obtained in three hours and seventeen minutes (with an Intel Pentium 4 processor, 2 GHz, 1 Go of RAM), which corresponds to a mean of twenty-five minutes per branch. One branch composed by a single segment was not extended. This can thus be considered as a false alarm and removed from the final tree. The extension result of three branches of the second tree was obtained in one hour and twenty minutes. The root branch (in red) was completely detected on step 1, and the convergence was obtained in one second. The two other branches were obtained in more than thirty minutes.

These results are very encouraging given the speckle noise present in this image and the weak contrast of the end of branches as shown by figure 13.

6.2 Generating new branches

The hierarchical modeling of the network allows to complete the partial network obtained in the initialization phase using a recursive algorithm that generates new branches from each detected branch c. This generation is

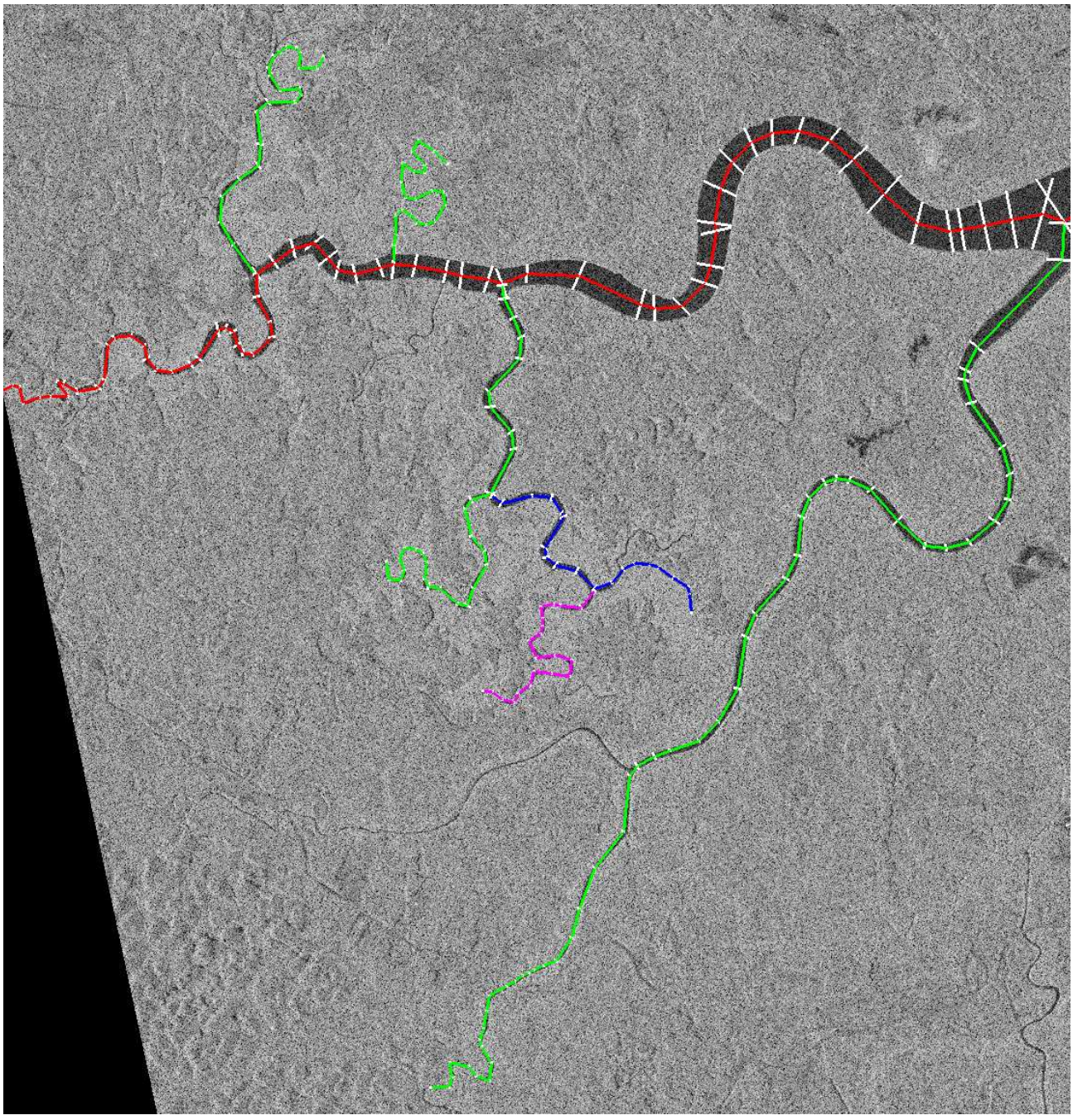


Figure 11: Extension of the first tree by simulated annealing using an RJMCMC algorithm.

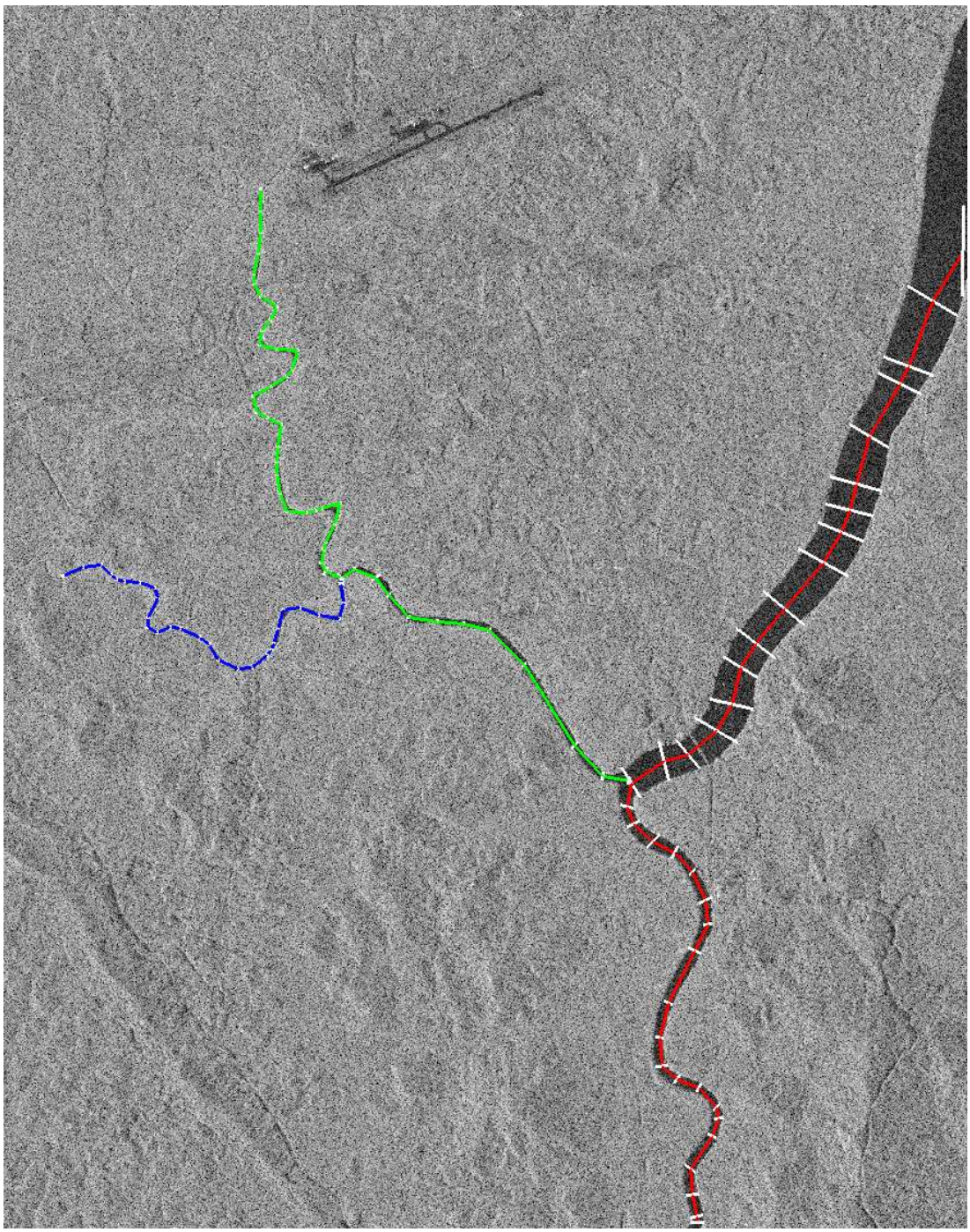


Figure 12: Extension of the second tree by simulated annealing using an RJMCMC algorithm.



data step 1 step 2

Figure 13: Zoom showing the results of the first step (connected components to polyline trees) and of the second step (polyline extension) on the end of a branch whose extraction is difficult: a weak contrast is added to the speckle of the radar satellite image.

based on the definition of a process in the neighborhood of an object as described in section 5.2.

Given all detected objects, we perform an optimization of the process associated to c, *i.e.* a maximization of the density given by:

$$h(X_c|\mathbf{C}) \propto h_d(X_c|\mathbf{C})\ h_p(X_c|\mathbf{C}) \tag{27}$$

The optimization is done via a simulated annealing using a Reversible Jump MCMC (RJMCMC) algorithm with an adaptive decrease of the temperature and an automatic convergence detection. We use three reversible moves: birth-and-death (BD) of a polyline, add/remove (AR) a segment to/from the end of a polyline; translation (T) of a point of a polyline.

The **birth-and-death move** corresponds to the proposition of the addition/removal of a polyline composed of a single segment. Indeed, this BD move combined to an AR move would be much more relevant than uniform birth-and-death in the entire state, since births of long polylines would almost always be rejected. This moves consists of the proposition of a birth with the probability p_b and the proposition of a death with the probability $p_d = 1 - p_b$. The birth of a polyline reduced to a segment is uniform: the proposed segment is uniformly drawn in $V_c \times [L_{min}, L_{max}] \times]\alpha_0 - \pi/2, \alpha_0 + \pi/2]$. The death is also uniform: a segment is uniformly chosen in the set $E_1(X_c)$ of polylines in X_c reduced to one segment. In the case of a birth, the Green ratio is given by:

$$R_{\mathrm{BD}}(X_c, X_c \cup x) = \frac{p_d}{p_b}\ \frac{\lambda |V_c|}{n_{max}\ (\sharp(E_1(X_c)) + 1)}\ \frac{h(X_c \cup x|\mathbf{C})}{h(X_c|\mathbf{C})} \tag{28}$$

where $\sharp(E_1(X_c))$ denotes the number of polylines in X_c reduced to one segment. Likewise, in the case of a death:

$$R_{\mathrm{BD}}(X_c, X_c \setminus x) = \frac{p_b}{p_d}\ \frac{n_{max}\ \sharp(E_1(X_c))}{\lambda V_c}\ \frac{h(X_c \setminus x|\mathbf{C})}{h(X_c|\mathbf{C})} \tag{29}$$

The two last moves - **add/remove** and **translation** - correspond to the ones proposed in the polyline extension procedure with the additional step of uniformly choosing a polyline among all generated polylines of X_c, and with no initial parameters (n_x is equal to 0). The Green ratios are given by equations (24-26).

The result of this algorithm applied to the initial tree given in figure 11 is presented in figure 14. It was obtained in 2 hours (with an Intel Pentium 4 processor, 3.4 GHz, 1 Go of RAM). The result is satisfactory as only one branch was not detected with respect to the manual extraction provided by the BRGM. Moreover, there are only two small false alarms. Only two branches are longer than the corresponding manually extracted branches, and one of these branches seems to be better detected by our algorithm than by the human operator. Two branches are not sufficiently extended due to the progressive contrast decrease.

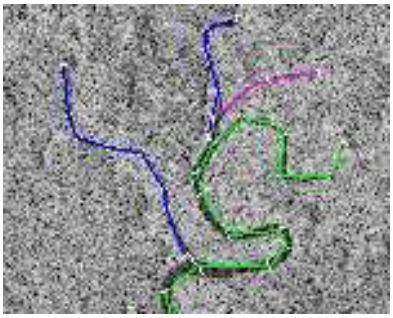
As it was expected no descendant was generated for the second tree (all branches were detected during the initialization phase).



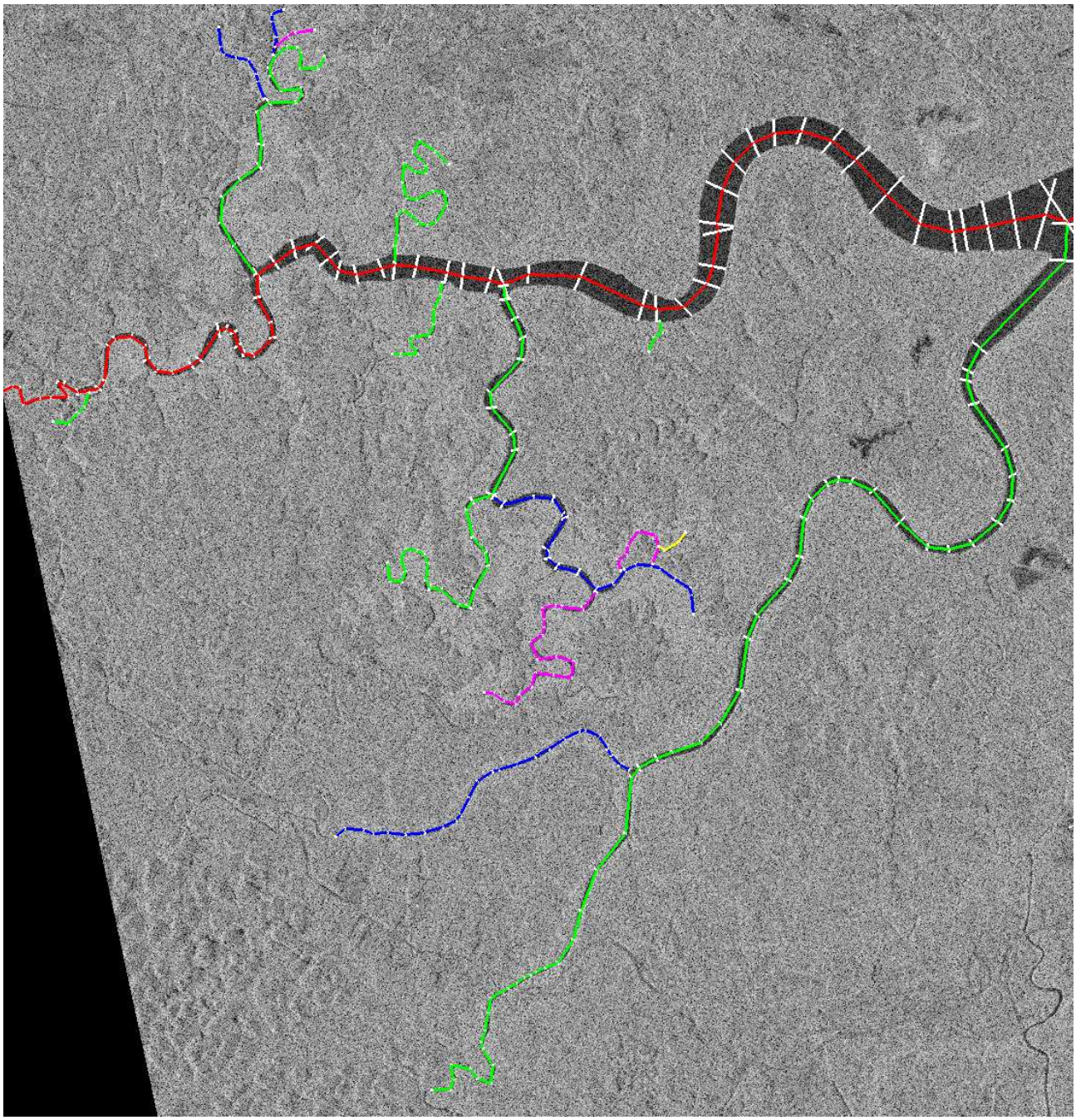
(a) data



(b) manual extraction



(c) automatic extraction



(d) result on tree 1

Figure 14: Branch generation from the first tree. The images (a), (b) and (c) correspond to a zoom on a particular zone where the automatically extracted network seems to be more exhaustive than the manually extracted network. The image (d) represents the final result.

7 Conclusion

We have proposed a method for unsupervised network extraction from satellite radar images combining the advantages of two approaches: a segmentation using an MRF and an object extraction using stochastic geometry. The MRF performs an efficient extraction in terms of computing time and in terms of detection of the surface part of the network. Nevertheless, the line part of the network is not detected with such an approach. This study has shown that the object processes bring a solution when the MRF approaches reach their limits. Indeed, this technique allows us to extract almost all rivers present in the scene. This is efficiently done thanks to the use of the segmentation result and the exploitation of the tree structure of hydrographic networks. Nevertheless, the extraction of the rivers near springs remains difficult due to a contrast decrease in direction of the spring. A solution could be to incorporate this information in the data model and to estimate the decrease law of the contrast along the polyline. Moreover, the proposed stochastic model should allow us to perform data fusion in order to benefit from the contribution of several sources (for instance, multi-sensor or mulX-Mozilla-Status: 8000 X-Mozilla-Status2: 00000000 ti-temporal data). Finally, this work could be extended to other types of networks with a tree structure such as the coronary trees in medical images.

Acknowledgments

The first three authors would like to thank the French Geological Survey (BRGM) for partial financial support, for providing remotely sensed images and reference networks. Part of this work has been conducted within the PNTS program.

References

[1] A. Baddeley and M. N. M. van Lieshout. Stochastic geometry models in high-level vision. *Statistics and Images*, 1:233–258, 1993.

[2] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 1975.

[3] J. Besag. On the statistical analysis of dirty pictures. *Journal of Royal Statistic Society*, B(68):259–302, 1986.

[4] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[5] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[6] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, 1987.

[7] X. Descombes, J.F. Mangin, E. Pechersky, and M.Sigelle. Fine structure preserving Markov model for image processing. In *9th Scandinavian Conference on Image Analysis*, pages 349–356, Uppsala, Suède, June 1995.

[8] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[9] T. Géraud. Fast road network extraction in satellite images using mathematical morphology and Markov random fields. In *IEEE - EURASIP Workshop on Nonlinear Signal and Image Processing*, June 2003.

[10] C. J. Geyer and J. Møller. Simulation and likelihood inference for spatial point process. *Scandinavian Journal of Statistics, Series B*, 21:359–373, 1994.

[11] C.J. Geyer. Likelihood inference for spatial point processes. In O.E. Barndorff-Nielsen, W.S. Kendall, and M.N.M van Lieshout, editors, *Stochastic Geometry: Likelihood and Computation*, chapter 3, pages 79–140. Chapman & Hall, 1999.

[12] P.J. Green. Reversible jump Markov chain Monte-Carlo computation and Bayesian model determination. *Biometrika*, 57:97–109, 1995.

[13] K.H. Hoffmann, D. Würtz, C. Groot, and M. Hanf. Concepts in optimizing simulated annealing schedules: an adaptive approach for parallel and vector machines. In M. Grauer and D.B. Pressmar, editors, *Parallel and Distributed Optimization*. Springer Verlag, 1991.

[14] M. Imberty and X. Descombes. Simulation de processus objets : Etude de faisabilité pour une application à la segmentation d'images. Research Report 3881, INRIA, Sophia Antipolis, France, 2000.

[15] C. Lacoste. *Extraction de Réseaux Linéiques à partir d'Images Satellitaires et Aériennes par Processus Ponctuels Marqués*. Phd thesis (in French), University of Nice - Sophia Antipolis, France, September 2004.

[16] C. Lacoste, X. Descombes, and J. Zerubia. Point processes for unsupervised line network extraction in remote sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1568–1579, October 2005.

[17] C. Lacoste, X. Descombes, J. Zerubia, and N. Baghdadi. Unsupervised line network extraction from remotely sensed images by polyline process. In *EUSIPCO*, Vienna, Austria, September 2004.

[18] C. Preston. Spatial birth and death processes. *Bulletin of the International Statistical Institute*, 46(2):371–391, 1976.

[19] B. D. Ripley. Modelling spatial patterns. *Journal of the Royal Statistical Institute, Series B*, 39:172–212, 1977.

[20] R. Stoica, X. Descombes, and J. Zerubia. A Gibbs point process for road extraction in remotely sensed images. *International Journal of Computer Vision*, 57(2):121–136, 2004.

[21] F. Tupin, H. Maitre, J-F. Mangin, J-M. Nicolas, and E. Pechersky. Detection of linear features in SAR images: Application to road network extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 36(2):434–453, 1998.

[22] M.N.M. van Lieshout. Stochastic annealing for nearest-neighbour point processes with application to object recognition. Research Report BS-R9306, CWI, Amsterdam, The Netherlands, 1993.

[23] M.N.M. van Lieshout. *Markov Point Processes and their Applications.* Imperial College Press, 2000.

[24] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: a Mathematical Introduction.* second edition, Springer-Verlag, 2003.



Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399